Approximate Nearest Neighbor Search for Low Dimensional Queries*

Sariel Har-Peled† Nirman Kumar‡

February 7, 2012

Abstract

We study the Approximate Nearest Neighbor problem for metric spaces where the query points are constrained to lie on a subspace of low doubling dimension.

1 Introduction

The nearest neighbor problem is the following. Given a set P of n data points in a metric space $\mathcal{X}$, preprocess P, such that given a query point $\mathsf{q} \in \mathcal{X}$, one can find (quickly) the point $\mathsf{n}_\mathsf{q} \in \mathsf{P}$ closest to q. Nearest neighbor search is a fundamental task used in numerous domains including machine learning, clustering, document retrieval, databases, statistics, and many others.

Exact nearest neighbor. The problem has a naive linear time algorithm without any preprocessing. However, by doing some nontrivial preprocessing, one can achieve a sublinear search time for the nearest neighbor. In d-dimensional Euclidean space (i.e., $\mathbb{R}^d$) this can be done by using Voronoi diagrams [dBCvKO08]. However, this approach is only suitable for low dimensions as the complexity of the Voronoi diagram is $O\big(n^{\lceil d/2 \rceil}\big)$. Specifically, Clarkson [Cla88] showed a data-structure with query time $O(\log n)$ time, and $O\big(n^{\lceil d/2 \rceil + \delta}\big)$ space, where $\delta > 0$ is a prespecified constant (the $O(\cdot)$ notation here hides constants that are exponential in the dimension). One can tradeoff the space used and the query time [AM93]. Meiser [Mei93] provided a data-structure with query time $O(d^5 \log n)$ (which has polynomial dependency on the dimension), where the space used is $O\big(n^{d+\delta}\big)$. These solutions are impractical even for data-sets of moderate size if the dimension is larger than two.

*Work on this paper was partially supported by a NSF AF award CCF-0915984.

†Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://www.uiuc.edu/~sariel/.

‡Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; nkumar5@illinois.edu; http://www.cs.uiuc.edu/~nkumar5/.

Approximate nearest neighbor. In typical applications, however, it is usually sufficient to return an ***approximate nearest neighbor*** (**ANN**). Given an $\varepsilon > 0$, a $(1+\varepsilon)$-ANN, to a query point q, is a point $y \in \mathsf{P}$, such that

$$d(\mathsf{q}, y) \leq (1+\varepsilon) d(\mathsf{q}, \mathsf{n}_{\mathsf{q}}) ,$$

where $\mathsf{n}_{\mathsf{q}} \in \mathsf{P}$ is the nearest neighbor to q in P. Considerable amount of work was done on this problem, see [Cla06] and references therein.

In high dimensional Euclidean space, Indyk and Motwani showed that ANN can be reduced to a small number of near neighbor queries [IM98]. Next, using locality sensitive hashing they provide a data-structure that answers ANN queries in time (roughly) $\tilde{O}\big(n^{1/(1+\varepsilon)}\big)$ and preprocessing time and space $\tilde{O}\big(n^{1+1/(1+\varepsilon)}\big)$. This was improved to $\tilde{O}\Big(n^{1/(1+\varepsilon)^2}\Big)$ query time, and preprocessing time and space $\tilde{O}\Big(n^{1+1/(1+\varepsilon)^2}\Big)$ [AI06, AI08]. These bounds are near optimal [MNP06].

In low dimensions (i.e., $\mathbb{R}^d$), one can use linear space (independent of ε) and get ANN query time $O(\log n + 1/\varepsilon^{d-1})$ [AMN+98, Har10]. Interestingly, for this data-structure, the approximation parameter ε is not prespecified during the construction; one needs to provide it only during the query. An alternative approach, is to use Approximate Voronoi Diagrams (AVD), introduced by Har-Peled [Har01], which are partition of space into regions, desirably of low complexity, typically with a representative point for each region that is an ANN for any point in the region. In particular, Har-Peled showed that there is such a decomposition of size $O\big((n/\varepsilon^d)\log^2 n\big)$, such that ANN queries can be answered in $O(\log n)$ time. Arya and Malamatos [AM02] showed how to build AVDs of linear complexity (i.e., $O(n/\varepsilon^d)$). Their construction uses Well Separated Pair Decompositions [CK95]. Further tradeoffs between query and space for AVDs were studied by Arya *et al.* [AMM09].

Metric spaces. One possible approach for the more general case, when the data lies in some abstract metric space, is to define a notion of dimension and develop efficient algorithms in these settings. This approach is motivated by the belief that real world data is "low dimensional" in many cases, and should be easier to handle than true high dimensional data. An example of this approach is the notion of ***doubling dimension*** [Ass83, Hei01, GKL03]. The ***doubling constant*** of metric space $\mathcal{X}$ is the maximum, over all balls b in the metric space $\mathcal{X}$, of the minimum number of balls needed to cover b, using balls with half the radius of b. The logarithm of the doubling constant is the ***doubling dimension*** of the space. The doubling dimension can be thought of as a generalization of the Euclidean dimension, as $\mathbb{R}^d$ has $\Theta(d)$ doubling dimension. Furthermore, the doubling dimension extends the notion of growth restricted metrics of Karger and Ruhl [KR02].

The problem of ANN in spaces of low doubling dimension was studied in [KR02, HKMR04]. Talwar [Tal04] presented several algorithms for spaces of low doubling dimension. Some of them were however dependent on the spread of the point set. Krauthgamer and Lee [KL04] presented a net navigation algorithm for ANN in spaces of low doubling dimension. Har-Peled and Mendel [HM06] provided data-structures for ANN search that use linear space and

match the bounds known for $\mathbb{R}^d$ [AMN+98]. Clarkson [Cla06] presents several algorithms for nearest neighbor search in low dimensional spaces for various notions of dimensions.

ANN in high and low dimensions. As indicated above, the ANN problem is easy in low dimensions (either Euclidean or bounded doubling dimension). If the dimension is high the problem is considerably more challenging. There is considerable work on ANN in high dimensional Euclidean space (see [IM98, KOR00]) but the query time is only slightly sublinear if ε is close to 0. In general metric spaces, it is easy to argue that (in the worst case) the ANN algorithm must compute the distance of the query point to all the input points.

It is natural to ask therefore what happens when the data (or the queries) come from a low dimensional subspace that lies inside a high dimensional ambient space. Such cases are interesting, as it is widely believed that in practice, real world data usually lies on a low dimensional manifold (or is close to lying on such manifold). Such low-dimensionality arises from the way the data is being acquired, inherent dependency between parameters, aggregation of data that leads to concentration of mass phenomena, etc.

Indyk and Naor [IN07] showed that if the data is in high dimensional Euclidean space, but lies on a manifold with low doubling dimension, then one can do a dimension reduction into constant dimension (i.e., similar in spirit to the JL lemma [JL84]), such that $(1+\varepsilon)$-ANN to a query point (the query point might lie anywhere in the ambient space) is preserved with constant probability. Using an appropriate data-structure on the embedded space and repeating this process sufficient number of times, results in a data-structure that can answer such ANN queries in polylog time (ignoring the dependency on ε).

The problem. In this paper, we study the "reverse" problem. Here we are given a high dimensional data set P, and we would like to preprocess it for ANN queries, where the queries come from a low-dimensional subspace/manifold $\mathcal{M}$. The question arises naturally when the given data is formed by a large number of data sets, while the ANN queries come from a single data set.

In particular, the meta question here is whether this problem is low or high dimensional in nature. Note, direct dimension reduction as done by Indyk and Naor would not work in this case. Indeed, imagine the data lies densely on a slightly deformed sphere in high dimensions, and the query is the center of the sphere. Clearly, a random dimension reduction into constant dimension would not preserve the $(1+\varepsilon)$-ANN (with high probability).

Our results. Given a point set P in a general metric space $\mathcal{X}$ (which is not necessarily Euclidean and is conceptually high dimensional), and a subspace $\mathcal{M}$ having low doubling dimension, we show how to preprocess P such that given any query point in $\mathcal{M}$ we can quickly answer $(1+\varepsilon)$-ANN queries on P. In particular, we get data-structures of (roughly) linear size that answer $(1+\varepsilon)$-ANN queries in (roughly) logarithmic time.

Our construction uses ideas developed for handling the low dimensional case. Initially, we embed P and $\mathcal{M}$ into a space with low doubling dimension that (roughly) preserves distances between $\mathcal{M}$ and P. We can use the embedded space to answer constant factor ANN queries.

Getting a better approximation requires some further ideas. In particular, we build a data-structure over $\mathcal{M}$ that is remotely similar to Approximate Voronoi Diagrams [Har01]. By sprinkling points carefully on the subspace $\mathcal{M}$ and using the net-tree data-structure [HM06] we can answer $(1+\varepsilon)$-ANN queries in time $O(\varepsilon^{-O(\dim)} + 2^{O(\dim)} \log n)$.

To get a better query time requires some further work. In particular, we borrow ideas from the simplified construction of Arya and Malamatos [AM02] (see also [AMM09]). Naively, this requires us to use well separated pairs decomposition (i.e., WSPD) [CK95] for P. Unfortunately, no such small WSPD exists for data in high dimensions. To overcome this problem, we build the WSPD in the embedded space. Next, we use this to guide us in the construction of the ANN data-structure. This results in a data-structure that can answer $(1+\varepsilon)$-ANN queries in $O(2^{O(\dim)} \log n)$ time. See Section 5 for details.

We also present an algorithm for a weaker model, where the query subspace is not given to us directly. Instead, every time an ANN query is issued, the algorithm computes a region around the query point such that the returned point is a valid ANN for all the points in the region. Furthermore, the algorithm caches such regions, and whenever a query arrives it first checks if the query point is already contained in one of the regions computed, and if so it answers the ANN query immediately. Significantly, for this algorithm we need no prespecified knowledge about the query subspace. The resulting algorithm computes on the fly AVD on the query subspace. In particular, we show that if the queries come from a subspace with doubling dimension dim then the algorithm would create at most $n/\varepsilon^{O(\dim)}$ regions overall. A restriction of this new algorithm is that we do not currently know how to efficiently perform a point-location query in a set of such regions, without assuming further knowledge about the subspace. Interestingly, the new algorithm can be interpreted as learning the underlying subspace/manifold the queries come from. See Section 6 for the precise result.

Organization. In Section 2, we define some basic concepts, and as a warm-up exercise study the problem where the subspace $\mathcal{M}$ is a linear subspace of $\mathbb{R}^d$ – this provides us with some intuition for the general case. We also present the embedding of P and $\mathcal{M}$ into the subspace $\mathcal{M}'$, which has low doubling dimension while (roughly) preserving distances of interest. In Section 3, we provide a data-structure for constant factor ANN using this embedding. In Section 4, we use the constant ANN to get a data-structure for answering $(1+\varepsilon)$-ANN. In Section 5, we use WSPD to build a data-structure that is similar in spirit to AVDs. This results in a data-structure with slightly faster ANN query time. The on the fly construction of AVD to answer ANN queries without assuming any knowledge of the query subspace is described in Section 6. Finally, conclusions are provided in Section 7.

2 Preliminaries

The Problem. We look at the ANN problem in the following setting. Given a set P of n data points in a metric space $\mathcal{X}$, and a set $\mathcal{M} \subseteq \mathcal{X}$ of (hopefully low) doubling dimension dim, and $\varepsilon > 0$, we want to preprocess the points of P, such that given a query point $\mathsf{q} \in \mathcal{M}$ one can efficiently find a $(1+\varepsilon)$-ANN of q in P.

Model. We are given a metric space $\mathcal{X}$ and a subset $\mathcal{M} \subseteq \mathcal{X}$ of doubling dimension dim. We assume that the distance between any pair of points can be computed in constant time in a black-box fashion. We also assume that one can build nets on $\mathcal{M}$. Specifically, given a point $p \in \mathcal{M}$ and a radius $r > 0$, we assume we can compute $2^{\dim}$ points $p_i \in \mathcal{M}$, such that $\mathsf{ball}(p, r) \cap \mathcal{M} \subseteq \bigcup \mathsf{ball}(p_i, r/2)$. By applying this recursively we can compute a ***r*-net** N for any $\mathsf{ball}(p, R)$ centered at p; that is, for any point $s \in \mathsf{ball}(p, R)$ there exists a point $u \in N$ such that $d(s, u) \leq r$. Let **compNet**(p, R, r) denote this algorithm for computing this r-net. The size of N is $(R/r)^{O(\dim)}$, and we assume this also bounds the time it takes to compute it.

Finally, given any point $p \in \mathcal{X}$ we assume that one can compute, in $O(1)$ time, a point $\alpha(p) \in \mathcal{M}$ such that $\alpha(p)$ is the closest point in $\mathcal{M}$ to p. (Alternatively, $\alpha(p)$ might be specified for each point of P in advance.)

Well separated pairs decomposition. For a point set P, a ***pair decomposition*** of P is a set of pairs $\mathcal{W} = \Big\{ \{A_1, B_1\}, \ldots, \{A_s, B_s\} \Big\}$, such that (I) $A_i, B_i \subset \mathsf{P}$ for every i, (II) $A_i \cap B_i = \emptyset$ for every i, and (III) $\cup_{i=1}^{s} A_i \otimes B_i = \mathsf{P} \otimes \mathsf{P}$.

A pair $\mathsf{Q} \subseteq \mathsf{P}$ and $\mathsf{R} \subseteq \mathsf{P}$ is ***(1/ε)-separated*** if $\max(\mathrm{diam}(\mathsf{Q}), \mathrm{diam}(\mathsf{R})) \leq \varepsilon \cdot d(\mathsf{Q}, \mathsf{R})$, where $d(\mathsf{Q}, \mathsf{R}) = \min_{p \in \mathsf{Q}, s \in \mathsf{R}} d(p, s)$. For a point set P, a ***well-separated pair decomposition*** (**WSPD**) of P with parameter $1/\varepsilon$ is a pair decomposition of P with a set of pairs $\mathcal{W} = \{\{A_1, B_1\}, \ldots, \{A_s, B_s\}\}$, such that, for any i, the sets A_i and B_i are ε^{-1}-separated [CK95].

2.1 Warm-up exercise: Affine Subspace.

We first consider the case where our query subspace is an affine subspace embedded in d dimensional Euclidean space. Thus let $\mathcal{X} = \mathbb{R}^d$ with the usual Euclidean metric. Suppose our query subspace $\mathcal{M}$ is an affine subspace of dimension k where $k \ll d$. We are also given n data points $\mathsf{P} = \{p_1, p_2, \ldots, p_n\}$. We want to preprocess P such that given a $\mathsf{q} \in \mathcal{M}$ we can quickly find a point $p_i \in \mathsf{P}$ which is a $(1+\varepsilon)$-**ANN** of q in P.

We choose an orthonormal system of coordinates for $\mathcal{M}$. Denote the projection of a point p to $\mathcal{M}$ as $\alpha(p)$. Denote the coordinates of a point $\alpha(p) \in \mathcal{M}$ in the chosen coordinate system as $(p^1, p^2, \ldots, p^k)$. Let $\mathsf{h}(p)$ denote the distance of a $p \in \mathbb{R}^d$ from the subspace $\mathcal{M}$. Notice that $\mathsf{h}(p) = \|p - \alpha(p)\|$. Consider the following embedding $p' = (p^1, p^2, \ldots, p^k, \mathsf{h}(p)) \in \mathbb{R}^{k+1}$.

It is easy to see that for $x \in \mathcal{M}$ and $y \in \mathbb{R}^d$, $\|x - y\|^2 = \|x - \alpha(y)\|^2 + \|\alpha(y) - y\|^2 = \|x - \alpha(y)\|^2 + \mathsf{h}(y)^2 = \|x' - y'\|^2$. So, $\|x - y\| = \|x' - y'\|$.

As such, if we can find a $(1+\varepsilon)$-**ANN** p'_i of q' in $\mathbb{R}^{k+1}$ then p_i is a $(1+\varepsilon)$-**ANN** of q. But this is easy to do using known data-structures for **ANN** [AMN+98], or the data-structures for approximate Voronoi diagram [Har01, AM02].

Thus, we have n points in $\mathbb{R}^{k+1}$ to preprocess and without loss of generality we can assume that p'_i are all distinct. Now given $\varepsilon \leq 1/2$, we can preprocess the points $\{p'_1, \ldots, p'_n\}$ and construct an approximate Voronoi diagram consisting of $O\big(n\varepsilon^{-(k+1)} \log \varepsilon^{-1}\big)$ regions.

Each such region is the difference of two cubes. Given a point $\mathsf{q}' \in \mathbb{R}^{k+1}$ we can find a $(1+\varepsilon)$-ANN in time $O(\log(n/\varepsilon))$.

2.2 An Embedding.

We show how to embed the points of P (and in fact all of $\mathcal{X}$) into another metric space $\mathcal{M}'$ with finite doubling dimension, such that the distances between P and $\mathcal{M}$ are roughly preserved.

For a point $p \in \mathcal{X}$, let $\alpha(p)$ denote the closest point in $\mathcal{M}$ to p (for the sake of simplicity of exposition we assume this point is unique). The ***height*** of a point $p \in \mathcal{X}$ is the distance between p and $\alpha(p)$; namely, $\mathsf{h}(p) = d(p, \alpha(p))$. Generalizing this, for a given set $A \subseteq \mathcal{X}$, we will let $\alpha(A)$ denote the set $\left\{ \alpha(x) \;\middle|\; x \in A \right\}$.

The metric space $\mathcal{M}'$ is $\mathcal{M} \times \mathbb{R}^+$. The embedding $\varphi : \mathcal{X} \to \mathcal{M}'$ maps a point $p \in \mathcal{X}$ into the point $\varphi(p) = (\alpha(p), \mathsf{h}(p))$. For a point $p \in \mathcal{X}$, we use $p' = \varphi(p)$ to denote the embedded point. The distance between any two points $p' = (\alpha(p), \mathsf{h}(p))$ and $s' = (\alpha(s), \mathsf{h}(s))$ of $\mathcal{M}'$ is defined as

$$d_{\mathcal{M}'}(p', s') = d_{\mathcal{M}'}\big(\alpha(p), \alpha(s)\big) + |\mathsf{h}(p) - \mathsf{h}(s)| \,.$$

It is easy to verify that $d_{\mathcal{M}'}(\cdot, \cdot)$ complies with the triangle inequality. For the sake of simplicity of exposition, we assume that for any two distinct points p and s in our (finite) input point set P it holds that $p' \neq s'$ (that is, $d_{\mathcal{M}'}(p', s') \neq 0$). This can be easily guaranteed by introducing symbolic perturbations.

Lemma 2.1 *The following holds: (A) For any two points $x, y \in \mathcal{M}$, we have $d_{\mathcal{M}'}(x', y') = d_{\mathcal{X}}(x, y)$. (B) For any point $x \in \mathcal{M}$ and $y \in \mathcal{X}$, we have $d_{\mathcal{X}}(x, y) \leq d_{\mathcal{M}'}(x', y') \leq 3 d_{\mathcal{X}}(x, y)$. (C) The metric space $\mathcal{M}'$ has doubling dimension at most $2 \dim + 2$.*

Proof: (A) Clearly, for $x, y \in \mathcal{M}$, we have $x' = (x, 0)$ and $y' = (y, 0)$. As such, $d_{\mathcal{M}'}(x', y') = d_{\mathcal{X}}(x, y) + |0 - 0| = d_{\mathcal{X}}(x, y)$.

(B) Let $x \in \mathcal{M}$ and $y \in \mathcal{X}$. We have $x' = (x, 0)$ and $y' = (\alpha(y), d_{\mathcal{X}}(y, \alpha(y)))$. As such,

$$\begin{aligned} d_{\mathcal{M}'}(x', y') &= d_{\mathcal{X}}(\alpha(x), \alpha(y)) + |0 - \mathsf{h}(y)| \\ &= d_{\mathcal{X}}(x, \alpha(y)) + d_{\mathcal{X}}(\alpha(y), y) \\ &\geq d_{\mathcal{X}}(x, y) \,, \end{aligned}$$

by the triangle inequality. On the other hand because $d_{\mathcal{X}}(y, \alpha(y)) \leq d_{\mathcal{X}}(y, x)$,

$$\begin{aligned} d_{\mathcal{M}'}(x', y') &= d_{\mathcal{X}}(x, \alpha(y)) + d_{\mathcal{X}}(y, \alpha(y)) \\ &\leq d_{\mathcal{X}}(x, y) + 2 d_{\mathcal{X}}(y, \alpha(y)) \\ &\leq 3 d_{\mathcal{X}}(x, y) \,, \end{aligned}$$

(C) Let (p, a) be a point in $\mathcal{M}'$ and consider the ball $\mathsf{b} = \mathsf{ball}_{\mathcal{M}'}((p, a), r) \subseteq \mathcal{M}'$ of radius r with center (p, a). Consider the projection of b into $\mathcal{M}$; that is $P_{\mathcal{M}} = \left\{ s \;\middle|\; (s, h) \in \mathsf{b} \right\}$. Similarly, let $P_{\mathbb{R}} = \left\{ h \;\middle|\; (s, h) \in \mathsf{b} \right\}$.

Clearly, $\mathsf{ball}_{\mathcal{M}'}((p,a),r) \subseteq P_{\mathcal{M}} \times P_{\mathbb{R}}$, and $\mathsf{P}_{\mathcal{M}}$ is contained in the ball $\mathsf{ball}_{\mathcal{M}}(p,r) = \mathsf{ball}_{\mathcal{X}}(p,r) \cap \mathcal{M}$. Since the doubling dimension of $\mathcal{M}$ is dim, this ball can be covered by $2^{2\dim}$ balls $\mathsf{ball}_{\mathcal{M}}(p_i, r/4)$ with centers $p_i \in \mathcal{M}$.

Also since $P_{\mathbb{R}} \subseteq \mathbb{R}$ is contained in an interval of length at most r, it can be covered by at most 4 intervals I_1, I_2, I_3, I_4 of length $r/4$ each, centered at values x_1, x_2, x_3, x_4, respectively. Then,

$$\begin{aligned}
\mathsf{ball}_{\mathcal{M}'}((p,a),r) &\subseteq P_{\mathcal{M}} \times P_{\mathbb{R}} \\
&\subseteq \bigcup_{j=1}^{4} \bigcup_{i} (\mathsf{ball}_{\mathcal{M}}(p_i, r/4) \cap \mathcal{M}) \times I_j \\
&\subseteq \bigcup_{j=1}^{4} \bigcup_{i} \mathsf{ball}_{\mathcal{M}'}\big((p_i, x_j), r/2\big)\,,
\end{aligned}$$

since the set $\mathsf{ball}_{\mathcal{M}}(p_i, r/4) \times I_j$ is contained in $\mathsf{ball}_{\mathcal{M}'}((p_i, x_j), r/2)$. We conclude that $\mathsf{ball}_{\mathcal{M}'}((p,a),r)$ can be covered using at most $2^{2\dim+2}$ balls of half the radius. ∎

3 A Constant Factor ANN Algorithm

In the preprocessing stage, we map the points of P into the metric space $\mathcal{M}'$ of Lemma 2.1. Build a net-tree for the point set $\mathsf{P}' = \left\{ p' \;\middle|\; p \in \mathsf{P} \right\}$ in $\mathcal{M}'$ and preprocess it for ANN queries using the data-structure of Har-Peled and Mendel [HM06]. Let $\mathcal{D}$ denote the resulting data-structure.

Answering a query. Given $\mathsf{q} \in \mathcal{M}$, we compute a 2-ANN to $\mathsf{q}' \in \mathcal{M}'$. Let this be the point y'. Return $d(\mathsf{q}, y)$.

Correctness. Let n_{q} be the nearest neighbor of q in P and y the point returned. We have,

$$\begin{aligned}
d_{\mathcal{M}'}\big(\mathsf{q}', \mathsf{n}'_{\mathsf{q}}\big) &= d_{\mathcal{X}}(\mathsf{q}, \alpha(\mathsf{n}_{\mathsf{q}})) + \mathsf{h}(\mathsf{n}_{\mathsf{q}}) \\
&\leq d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_{\mathsf{q}}) + \mathsf{h}(\mathsf{n}_{\mathsf{q}}) + \mathsf{h}(\mathsf{n}_{\mathsf{q}}) \\
&\leq 3 d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_{\mathsf{q}})
\end{aligned}$$

As y' is a 2-ANN for q' and $\mathsf{q} \in \mathcal{M}$, we have

$$d_{\mathcal{X}}(\mathsf{q}, y) \leq d_{\mathcal{M}'}(\mathsf{q}', y') \leq 2 d_{\mathcal{M}'}\big(\mathsf{q}', \mathsf{n}'_{\mathsf{q}}\big) \leq 6 d(\mathsf{q}, \mathsf{n}_{\mathsf{q}})\,.$$

We thus proved the following.

Lemma 3.1 *Given a set $\mathsf{P} \subseteq \mathcal{X}$ of n points and a subspace $\mathcal{M}$ of doubling dimension* dim*, one can build a data-structure in $2^{O(\dim)} n \log n$ expected time, such that given a query point $\mathsf{q} \in \mathcal{M}$, one can return a 6-ANN to q in P in $2^{O(\dim)} \log n$ query time. The space used by this data-structure is $2^{O(\dim)} n$.*

Proof: Since the doubling dimension of $\mathcal{M}'$ is at most $2\dim+2$, building the net tree and preprocessing it for ANN queries takes $2^{O(\dim)}n\log n$ expected time, and the space used is $2^{O(\dim)}n$ [HM06]. The 2-ANN query for a point q takes time $2^{O(\dim)}\log n$. ∎

4 Answering $(1+\varepsilon)$-ANN

Once we have a constant factor approximation to the nearest-neighbor in P it is not too hard to boost it into $(1+\varepsilon)$-ANN. To this end, we need to understand what the net-tree [HM06] provides us with. The following is implied by fiddling with the ANN algorithm of [HM06].

Lemma 4.1 *Given a net-tree for a set $\mathsf{Q}\subseteq\mathcal{M}$ of n points in a metric space with doubling dimension* dim*, and given a point $p\in\mathcal{M}$ and radii $r\leq R$, one can compute a r-net N of Q, such that the following properties hold:*

(A) For any point $s\in\mathsf{Q}\cap\mathsf{ball}(p,R)$ there exists a point $u\in N$ such that $d(s,u)\leq r$.
(B) $|N|=(R/r)^{O(\dim)}$.
(C) Each point of $p\in N$ corresponds to a node $v(p)$ in the net-tree. Let $\mathsf{Q}_{v(p)}$ denote the subset of points of Q stored in the subtree of $v(p)$. The union $\bigcup_{p\in N}\mathsf{Q}_{v(p)}$ covers $\mathsf{Q}\cap\mathsf{ball}(p,R)$.
(D) For any $p\in N$, the diameter of the point set $\mathsf{Q}_{v(p)}$ is bounded by r.
(E) The time to compute N is $2^{O(\dim)}\log n+O(|N|)$.

Construction. For every point $p\in\mathsf{P}$ we compute a $r(p)$-net $U(p)$ for $\mathsf{ball}_{\mathcal{M}}(\alpha(p),R(p))$, where $r(p)=\varepsilon\mathsf{h}(p)/(20c_1)$ and $R(p)=c_1\mathsf{h}(p)/\varepsilon$. Here c_1 is some sufficiently large constant. This net is computed using the algorithm **compNet**, see Section 2. This takes $1/\varepsilon^{O(\dim)}$ time to compute for each point of P.

For each point u of the net $U(p)\subseteq\mathcal{M}$ store the original point p it arises from, and the distance to the original point p. We will refer to $\mathsf{s}(u)=d(u,p)$ as the ***reach*** of u.

Let $\mathsf{Q}\subseteq\mathcal{M}$ be union of all these nets. Clearly, we have that $|\mathsf{Q}|=n/\varepsilon^{O(\dim)}$. Build a net-tree $\mathcal{T}$ for the points of Q. We compute in a bottom-up fashion for each node v of the net-tree $\mathcal{T}$ the point with the smallest reach stored in Q_v.

Answering a query. Given a query point $\mathsf{q}\in\mathcal{M}$, compute using the algorithm of Lemma 3.1 a 6-ANN to q in P. Let Δ be the distance from q to this ANN. Let $R=20\Delta$, and $r'=\varepsilon\Delta/20$. Using $\mathcal{T}$ and Lemma 4.1, compute a r'-net N of $\mathsf{ball}_{\mathcal{M}}(\mathsf{q},R)$.

Next, for each point of $p\in N$ consider its corresponding node $v(p)\in\mathcal{T}$. Each such node stores a point of minimum reach in $\mathsf{Q}_{v(p)}$. We compute the distance to each such minimum-reach point and return the nearest-neighbor found as the ANN.

Theorem 4.2 *Given a set $\mathsf{P}\subseteq\mathcal{X}$ of n points and a subspace $\mathcal{M}$ of doubling dimension* dim*, and a parameter $\varepsilon>0$, one can build a data-structure in $n\varepsilon^{-O(\dim)}\log n$ expected time, such that given a query point $\mathsf{q}\in\mathcal{M}$, one can return a $(1+\varepsilon)$-ANN to q in P in $2^{O(\dim)}\log n+\varepsilon^{-O(\dim)}$ query time.*

This data-structure uses $n\varepsilon^{-O(\dim)}$ space.

Proof: We only need to prove the bound on the quality of the approximation. Consider the nearest-neighbor $\mathsf{n_q}$ to q in P.

(A) If there is a point $z \in U(\mathsf{n_q}) \subseteq \mathsf{Q}$ in distance at most r' from q then there is a net point u of N that contains z in its subtree of $\mathcal{T}$. Let w_y be the point of minimum reach in $\mathsf{Q}_{v(u)}$, and let $y \in \mathsf{P}$ be the corresponding original point. Now, we have

$$d(\mathsf{q}, y) \leq d(\mathsf{q}, w_y) + d(w_y, y) \leq d(\mathsf{q}, w_y) + d(z, \mathsf{n_q})$$

as the point w_y has reach $d(w_y, y)$, w_y is the point of minimal reach among all the points of $\mathsf{Q}_{v(u)}$, $z \in \mathsf{Q}_{v(u)}$, and $d(z, \mathsf{n_q})$ is the reach of z. So, by the triangle inequality, we have

$$\begin{aligned} d(\mathsf{q}, y) &\leq d(\mathsf{q}, w_y) + d(\mathsf{q}, \mathsf{n_q}) + d(z, \mathsf{q}) \\ &\leq \big(d(\mathsf{q}, z) + d(z, w_y)\big) + d(\mathsf{q}, \mathsf{n_q}) + d(z, \mathsf{q}) \\ &\leq d(\mathsf{q}, \mathsf{n_q}) + 3r', \end{aligned}$$

as $z, w_y \in \mathsf{Q}_{v(u)}$ and the diameter of $\mathsf{Q}_{v(u)}$ is at most r'. So we have,

$$d(\mathsf{q}, y) \leq d(\mathsf{q}, \mathsf{n_q}) + 3\varepsilon\Delta/20 \leq (1+\varepsilon) d(\mathsf{q}, \mathsf{n_q}) \,.$$

(B) Otherwise, it must be that, $d(\mathsf{q}, U(\mathsf{n_q})) > r'$. Observe, that it must be that $r(\mathsf{n_q}) < r'$ as $\mathsf{h}(\mathsf{n_q}) \leq \Delta$. It must be therefore that the query point is outside the region covered by the net $U(\mathsf{n_q})$. As such, we have

$$\begin{aligned} R(\mathsf{n_q}) &= \frac{c_1 \mathsf{h}(\mathsf{n_q})}{\varepsilon} \\ &< d(\alpha(\mathsf{n_q}), \mathsf{q}) \\ &\leq d(\mathsf{q}, \mathsf{n_q}) + d(\mathsf{n_q}, \alpha(\mathsf{n_q})) \\ &\leq 2d(\mathsf{n_q}, \mathsf{q}) \leq 2\Delta, \end{aligned}$$

which means $\mathsf{h}(\mathsf{n_q}) \leq 2\varepsilon\Delta/c_1$. Namely, the height of the point $\mathsf{n_q}$ is insignificant in comparison to its distance from q (and conceptually can be considered to be zero). In particular, consider the net point $u \in N$ that contains $\alpha(\mathsf{n_q})$ in its subtree. The point of smallest reach in this subtree provides an $(1+\varepsilon)$-ANN as an easy but tedious argument similar to the one above shows. ∎

5 Answering $(1+\varepsilon)$-ANN faster

In this section, we extend the approach used in the above construction to get a data-structure which is similar in spirit to an AVD of P on $\mathcal{M}$. Specifically, we spread a set of points $\mathcal{C}$ on $\mathcal{M}$, and we associate a point of P with each one of them. Now, answering 2-ANN on $\mathcal{C}$, and returning the point of P associated with this point, results in the desired $(1+\varepsilon)$-ANN.

algBuildANN$(\mathsf{P}, \mathcal{M})$.

$\mathsf{P}' = \left\{ x' \;\middle|\; x \in \mathsf{P} \right\}$

Compute a 8-WSPD $\mathcal{W} = \{\{A'_1, B'_1\}, \ldots, \{A'_s, B'_s\}\}$ of P'

for $\{A'_i, B'_i\} \in \mathcal{W}$ **do**

Choose points $a'_i \in A'_i$ and $b'_i \in B'_i$.

$t_i = d_{\mathcal{M}'}(a'_i, b'_i), \quad T_i = t_i + \mathsf{h}_{\max}(A'_i) + \mathsf{h}_{\max}(B'_i)$

$R_i = c_2 T_i/\varepsilon, \quad r_i = \varepsilon T_i / c_2$

$N_i = \textbf{compNet}(\alpha(a_i), R_i, r_i) \cup \textbf{compNet}(\alpha(b_i), R_i, r_i).$

$\mathcal{C} = N_1 \cup \ldots \cup N_s$

$\mathcal{N}_{\mathcal{C}} \leftarrow$ Net-tree for $\mathcal{C}$ [HM06]

for $p \in \mathcal{C}$ **do**

Compute $\mathsf{nn}(p, \mathsf{P})$ and store it with p

Figure 1: Preprocessing the subspace $\mathcal{M}$ to answer $(1+\varepsilon)$-ANN queries on P. Here c_2 is a sufficiently large constant.

algANN ($\mathsf{q} \in \mathcal{M}$)

$p \leftarrow$ 2-ANN of q among $\mathcal{C}$

(Use net-tree $\mathcal{N}_{\mathcal{C}}$ [HM06] to compute p.)

return the point in P associated with p.

Figure 2: Find a $(1+O(\varepsilon))$-ANN in P for a query point $\mathsf{q} \in \mathcal{M}$.

5.1 The construction.

For a set $Z' \subseteq \mathsf{P}'$ let

$$\mathsf{h}_{\max}(Z') = \max_{(p,h) \in Z'} h.$$

The preprocessing stage is presented in Figure 1, and the algorithm for finding the $(1+\varepsilon)$-ANN for a given query is presented in Figure 2.

5.2 Analysis.

Suppose the data-structure returned y and the actual nearest neighbor of q is n_{q}. If $y = \mathsf{n}_{\mathsf{q}}$ then the algorithm returned the exact nearest-neighbor to q and we are done. Otherwise, by our general position assumption, we can assume that $y' \neq \mathsf{n}'_{\mathsf{q}}$.

Note, there is a WSPD pair $\{A', B'\} \in \mathcal{W}$ that separates y' from n'_{q} in $\mathcal{M}'$; namely, $y' \in A'$ and $\mathsf{n}'_{\mathsf{q}} \in B'$.

Let $t = d_{\mathcal{M}'}(a', b')$, where a' and b' are the representative points of A' and B', respectively. Now, let $T = \mathsf{h}_{\max}(A') + \mathsf{h}_{\max}(B') + t$, $R = c_2 T/\varepsilon$ and $r = \varepsilon T / c_2$.

Lemma 5.1 *If* $\mathsf{q} \notin \mathsf{ball}(\alpha(a), R) \cup \mathsf{ball}(\alpha(b), R)$ *then the algorithm returns* $(1+\varepsilon)$-*ANN in* P *to the query point* q *(assuming* $\mathfrak{c}_2$ *is sufficient large.).*

Proof: Observe that $d(\alpha(\mathsf{n}_\mathsf{q}), \alpha(y)) \leq d_{\mathcal{M}'}\left(\mathsf{n}'_\mathsf{q}, y'\right) \leq d_{\mathcal{M}'}(a', b') + \operatorname{diam}(A') + \operatorname{diam}(B') \leq t(1 + 1/8 + 1/8) = 5t/4$ by the 8-WSPD separation property. So, by the triangle inequality, we have $d_{\mathcal{X}}(\mathsf{n}_\mathsf{q}, y) \leq \mathsf{h}(\mathsf{n}_\mathsf{q}) + d(\alpha(\mathsf{n}_\mathsf{q}), \alpha(y)) + \mathsf{h}(y) \leq \mathsf{h}_{\max}(A') + (5/4)t + \mathsf{h}_{\max}(B') \leq (5/4)T$.

Since $\mathsf{n}'_\mathsf{q}, b' \in B'$, we have $d_{\mathcal{X}}(\alpha(\mathsf{n}_\mathsf{q}), \alpha(b)) \leq d_{\mathcal{M}'}\left(\mathsf{n}'_\mathsf{q}, b'\right) \leq \operatorname{diam}(B') \leq t/8 \leq T/8$. Therefore,

$$\begin{aligned} d_{\mathcal{X}}(\mathsf{q}, \alpha(\mathsf{n}_\mathsf{q})) &\geq d_{\mathcal{X}}(\mathsf{q}, \alpha(b)) - d_{\mathcal{X}}(\alpha(\mathsf{n}_\mathsf{q}), \alpha(b)) \\ &\geq \mathfrak{c}_2 \frac{T}{\varepsilon} - \operatorname{diam}(B') \\ &\geq T\left(\frac{\mathfrak{c}_2}{\varepsilon} - \frac{1}{8}\right) \\ &\geq \frac{\mathfrak{c}_2 T}{2\varepsilon}, \end{aligned}$$

assuming $\varepsilon \leq 1$ and $\mathfrak{c}_2 \geq 1$. Now, $d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) \geq d_{\mathcal{X}}(\mathsf{n}_\mathsf{q}, \alpha(\mathsf{n}_\mathsf{q}))$, and thus by the triangle inequality, we have

$$\begin{aligned} d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) &\geq \frac{d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) + d_{\mathcal{X}}(\mathsf{n}_\mathsf{q}, \alpha(\mathsf{n}_\mathsf{q}))}{2} \\ &\geq \frac{d_{\mathcal{X}}(\mathsf{q}, \alpha(\mathsf{n}_\mathsf{q}))}{2} \\ &\geq \frac{\mathfrak{c}_2 T}{4\varepsilon}. \end{aligned}$$

This implies that $d_{\mathcal{X}}(\mathsf{q}, y) \leq d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) + d_{\mathcal{X}}(\mathsf{n}_\mathsf{q}, y) \leq d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) + (5/4)T \leq (1+\varepsilon) d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q})$, assuming $\mathfrak{c}_2 \geq 5$. ∎

Lemma 5.2 *If* $\mathsf{q} \in \mathsf{ball}(\alpha(a), \mathfrak{c}_2 T/\varepsilon) \cup \mathsf{ball}(\alpha(b), \mathfrak{c}_2 T/\varepsilon)$ *then the algorithm returns* $(1+\varepsilon)$-*ANN in* P *to the query point* q.

Proof: Since the algorithm covered the set $\mathsf{ball}(\alpha(a), T/\varepsilon) \cup \mathsf{ball}(\alpha(b), T/\varepsilon)$ with a net of radius $r = \varepsilon T/\mathfrak{c}_2$, it follows that $d_{\mathcal{X}}(\mathsf{q}, \mathcal{C}) \leq r$. Let $\overline{\mathsf{c}}$ be the point in the 2-ANN search to q in $\mathcal{N}_{\mathcal{C}}$. We have $d_{\mathcal{X}}(\mathsf{q}, \overline{\mathsf{c}}) \leq 2r$. Now, the algorithm returned the nearest neighbor to $\overline{\mathsf{c}}$ as the ANN; that is, y is the nearest neighbor of $\overline{\mathsf{c}}$ in P.

If $d_{\mathcal{X}}(\mathsf{q}, y) \geq T/40$ then

$$\begin{aligned} d_{\mathcal{X}}(\mathsf{q}, \mathsf{n}_\mathsf{q}) &\geq d_{\mathcal{X}}(\overline{\mathsf{c}}, \mathsf{n}_\mathsf{q}) - d_{\mathcal{X}}(\mathsf{q}, \overline{\mathsf{c}}) \\ &\geq d_{\mathcal{X}}(\overline{\mathsf{c}}, y) - d_{\mathcal{X}}(\mathsf{q}, \overline{\mathsf{c}}) \\ &\geq (d_{\mathcal{X}}(\mathsf{q}, y) - d_{\mathcal{X}}(\mathsf{q}, \overline{\mathsf{c}})) - d_{\mathcal{X}}(\mathsf{q}, \overline{\mathsf{c}}) \\ &\geq d_{\mathcal{X}}(\mathsf{q}, y) - 4r \\ &= d_{\mathcal{X}}(\mathsf{q}, y) - 4\frac{\varepsilon T}{\mathfrak{c}_2} \\ &\geq (1 - \varepsilon/2) d_{\mathcal{X}}(\mathsf{q}, y), \end{aligned}$$

by the triangle inequality and if $c_2 \geq 320$. Since $1/(1-\varepsilon/2) \leq 1+\varepsilon$, we have that $d_{\mathcal{X}}(\mathsf{q}, y) \leq (1+\varepsilon) d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q})$.

If $d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}) \geq T/40$ then using similar argumentation to the above, we have that

$$\begin{aligned} d_{\mathcal{X}}(\mathsf{q}, y) &\leq d_{\mathcal{X}}(\overline{\mathrm{c}}, y) + d_{\mathcal{X}}(\mathsf{q}, \overline{\mathrm{c}}) \\ &\leq d_{\mathcal{X}}(\overline{\mathrm{c}}, y) + 2r \\ &\leq d_{\mathcal{X}}(\overline{\mathrm{c}}, \mathsf{n_q}) + 2r \\ &\leq d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}) + 4r \\ &= d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}) + 4\frac{\varepsilon T}{c_2} \\ &\leq (1+\varepsilon) d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}), \end{aligned}$$

assuming $c_2 \geq 160$.

If $d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}) \leq T/40$ and $d_{\mathcal{X}}(\mathsf{q}, y) \leq T/40$ then $\mathsf{h}(\mathsf{n_q}) \leq d_{\mathcal{X}}(\mathsf{q}, \mathsf{n_q}) \leq T/40$ and $\mathsf{h}(y) \leq d_{\mathcal{X}}(\mathsf{q}, y) \leq T/40$. Observe that

$$\mathsf{h}_{\max}(A') \leq \mathsf{h}(y) + \operatorname{diam}(A') \leq T/40 + \frac{t}{8} \leq \frac{3T}{20}.$$

and similarly $\mathsf{h}_{\max}(B') \leq 3T/20$. This implies that

$$\begin{aligned} (3/4)t &= t\left(1 - \frac{1}{8} - \frac{1}{8}\right) \\ &\leq d_{\mathcal{M}'}(a', b') - \operatorname{diam}(A') - \operatorname{diam}(B') \\ &\leq d_{\mathcal{M}'}\left(\mathsf{n}'_{\mathsf{q}}, y'\right) \\ &= |\mathsf{h}(\mathsf{n_q}) - \mathsf{h}(y)| + d_{\mathcal{X}}(\alpha(\mathsf{n_q}), \alpha(y)) \\ &\leq T/40 + d_{\mathcal{X}}(\alpha(\mathsf{n_q}), \mathsf{n_q}) + d_{\mathcal{X}}(\mathsf{n_q}, y) \\ &+ d_{\mathcal{X}}(y, \alpha(y)) \\ &\leq T/40 + \mathsf{h}(\mathsf{n_q}) \\ &+ (d_{\mathcal{X}}(\mathsf{n_q}, \mathsf{q}) + d_{\mathcal{X}}(\mathsf{q}, y)) + \mathsf{h}(y) \\ &\leq T/40 + 3T/20 + T/40 + T/40 + 3T/20 \\ &\leq 3T/8 \end{aligned}$$

This implies that $t \leq T/2$ and thus $T = t + \mathsf{h}_{\max}(A') + \mathsf{h}_{\max}(B') \leq T/2 + 3T/20 + 3T/20 = (4/5)T$. This implies that $T \leq 0$. We conclude that $d_{\mathcal{M}'}(a', b') = t \leq T \leq 0$. That implies that $a' = b'$, which is impossible, as no two points of P get mapped to the same point in $\mathcal{M}'$. (And of course, no point can appear in both sides of a pair in the WSPD.) ∎

The preprocessing time of the above algorithm is dominated by the task of computing for each point of $\mathcal{C}$ its nearest neighbor in P. Observe, that the algorithm would work even if we only use $(1+O(\varepsilon))$-ANN. Using Theorem 4.2 to answer these queries, we get the following result.

Theorem 5.3 *Given a set of* $\mathsf{P} \subseteq \mathcal{X}$ *of* n *points, and a subspace* $\mathcal{M}$ *of doubling dimension* dim*, one can construct a data structure requiring space* $n\varepsilon^{-O(\dim)}$*, such that given a query point* $\mathsf{q} \in \mathcal{M}$ *one can find a* $(1+\varepsilon)$*-ANN to* q *in* P*. The query time is* $2^{O(\dim)} \log n$*.*

The preprocessing time to build this data-structure is $n\varepsilon^{-O(\dim)} \log n$*.*

6 Online ANN

The algorithms of Section 4 and Section 5 require that the subspace of the query points is known, in that we can compute the closest point $\alpha(p)$ on $\mathcal{M}$ given a $p \in \mathcal{X}$, and that we can find a net for a ball on $\mathcal{M}$ using **compNet**, see Section 2. In this section we show that if we are able to efficiently answer membership queries in regions that are the difference of two balls, then, in fact, we do not require such explicit knowledge of $\mathcal{M}$. We construct an AVD on $\mathcal{M}$ in an online manner as the query points arrive. When a new query point arrives, we test for membership among the existing regions of the AVD. If a region contains the point we immediately output its associated ANN that is already stored with the region. Otherwise we use an appropriate algorithm to find a nearest neighbor for the query point and add a new region to the AVD.

Here we present our algorithm to compute the AVD in this online setting and prove that when the query points come from a subspace of low doubling dimension, the number of regions created is linear.

6.1 Online AVD Construction and ANN Queries.

The algorithm **algBuildAVD**$(\mathsf{P}, \mathcal{R}, \mathsf{q})$ is presented in Figure 3. The algorithm maintains a set of regions $\mathcal{R}$ that represent the partially constructed AVD. Given a query point q it returns an ANN from P and sometimes adds a region $\mathcal{R}_{\mathsf{q}}$ to $\mathcal{R}$. The quantity D' is a 2-approximation to the diameter D of P, and can be precomputed in $O(n)$ time. Let p be a fixed point of P.

The regions created by the algorithm in Figure 3 are the difference of two balls. An example region when the balls $\mathsf{ball}(\mathsf{q}, \varepsilon r_2/5)$ and $\mathsf{ball}(y, 4t/5\varepsilon)$ intersect is shown in Figure 4. The intuition as to why y is a valid ANN inside this region is as follows. Since the distance of q to y is r_1, the points inside $\mathsf{ball}(y, \varepsilon r_1/3)$ are all roughly the same distance from q. The next distance of interest is the closest point outside this ball. As long as we are inside $\mathsf{ball}(\mathsf{q}, \varepsilon r_2/5)$ the points outside $\mathsf{ball}(y, \varepsilon r_1/3)$ are too far and cannot be a $(1+\varepsilon)$-ANN. But if we get too close to y we can no more be certain that y is a valid $(1+\varepsilon)$-ANN, as it is no more true that distances to points inside $\mathsf{ball}(y, \varepsilon r_1/3)$ look all roughly the same. In other words, there may be points much closer than y, when we are close enough to y. Thus in a small enough neighborhood around y we need to zoom in and possibly create a new region. The formal proof of correctness follows from the following lemmas.

Lemma 6.1 *If* $d(\mathsf{q}, p) \geq 2\mathsf{D}' + 2\mathsf{D}'/\varepsilon$ *then* p *is a valid* $(1+\varepsilon)$*-ANN.*

algBuildAVD$(\mathsf{P}, \mathcal{R}, \mathsf{q})$.

comment: p is a fixed point in P.D′ is a 2-approximation to diam(P)

if $d(\mathsf{q}, p) \geq 2\mathsf{D}' + 2\mathsf{D}'/\varepsilon$ return p.

if $\exists \mathcal{R} \in \mathcal{R}$ with $\mathsf{q} \in \mathcal{R}$

return point y associated with $\mathcal{R}$.

Compute $(1+\varepsilon/10)$-ANN y of q in P. Let $r_1 = d(\mathsf{q}, y)$.

Let $s \in \mathsf{P}$ be the furthest point from y inside $\mathsf{ball}(y, \varepsilon r_1/3)$. Let $t = d(y, s)$.

if there is no point in $\mathsf{P} \setminus \mathsf{ball}(y, \varepsilon r_1/3)$.

Let $\mathcal{R}_\mathsf{q} = \mathsf{ball}(\mathsf{q}, \mathsf{D}'/4)$.

else

Compute $(1+\varepsilon/10)$-ANN $\bar{y}$ of q in $\mathsf{P} \setminus \mathsf{ball}(y, \varepsilon r_1/3)$. Let $r_2 = d(\mathsf{q}, \bar{y})$.

$\mathcal{R}_\mathsf{q} = \mathsf{ball}(\mathsf{q}, \varepsilon r_2/5) \setminus \mathsf{ball}(y, 5t/4\varepsilon)$.

$\mathcal{R} = \mathcal{R} \cup \mathcal{R}_\mathsf{q}$. Associate y with $\mathcal{R}_\mathsf{q}$.

return y as ANN for q.

Figure 3: Answering $(1+\varepsilon)$-ANN and constructing AVD

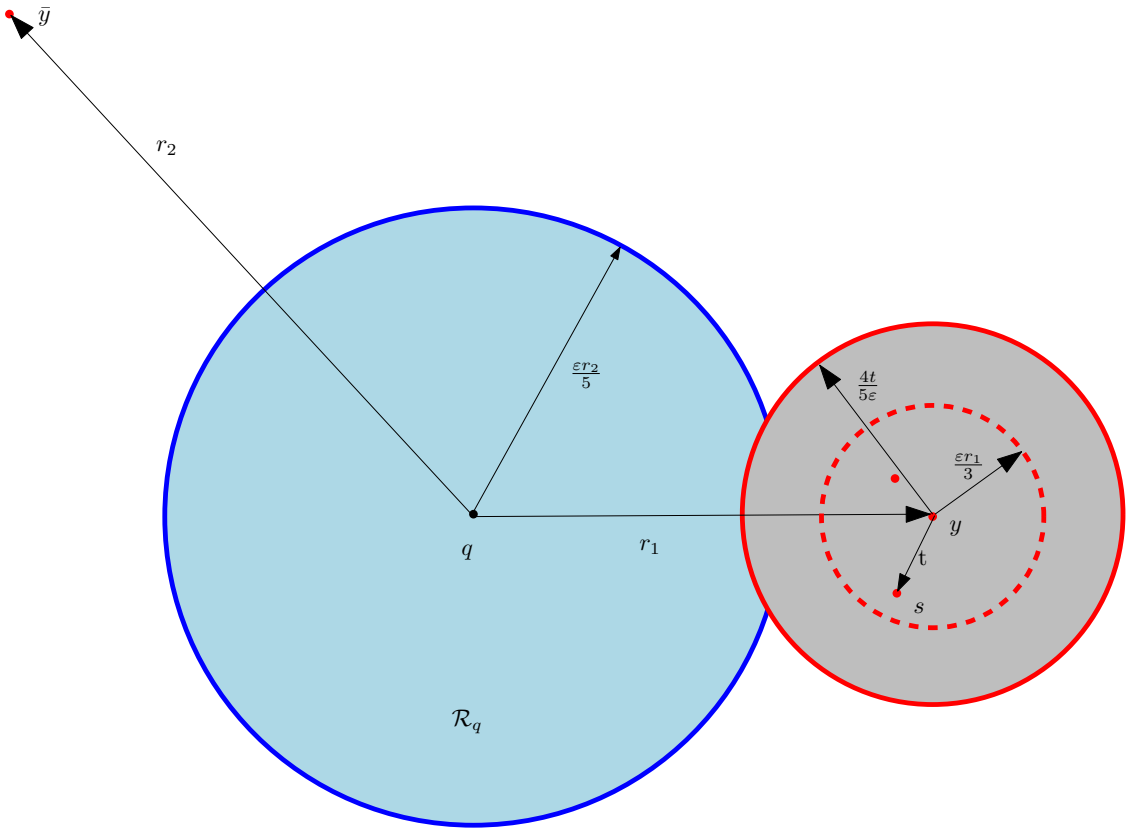

Figure 4: An example AVD region $\mathcal{R}_\mathsf{q}$

Proof: Since D' is a 2-approximation to the diameter of P, so $\mathsf{D} \leq 2\mathsf{D}'$. This means $d(\mathsf{q}, p) \geq \mathsf{D} + \mathsf{D}/\varepsilon$. Let $\mathsf{n}_\mathsf{q} \in \mathsf{P}$ be the closest point to q. By the triangle inequality,

$$\mathsf{D} + \mathsf{D}/\varepsilon \leq d(\mathsf{q}, p) \leq d(\mathsf{q}, \mathsf{n}_\mathsf{q}) + d(\mathsf{n}_\mathsf{q}, p) \leq d(\mathsf{q}, \mathsf{n}_\mathsf{q}) + \mathsf{D}.$$

This together with $d(\mathsf{q}, p) \leq d(\mathsf{q}, \mathsf{n}_\mathsf{q}) + \mathsf{D}$ implies that $d(\mathsf{q}, p) \leq (1+\varepsilon) d(\mathsf{q}, \mathsf{n}_\mathsf{q})$. ∎

Lemma 6.2 *If there is no region $\mathcal{R}$ containing q the algorithm outputs a valid $(1+\varepsilon)$-ANN.*

Proof: We output y which is a $(1+\varepsilon/10)$-ANN of q. ∎

The next lemma finally completes the argument.

Lemma 6.3 *The $(1+\varepsilon/10)$-ANN y found in the algorithm is an $(1+\varepsilon)$-ANN for any point $\bar{\mathsf{q}} \in \mathcal{R}_\mathsf{q}$ constructed in the algorithm.*

Proof: There are two possibilities.

(A) If the region $\mathcal{R}_\mathsf{q}$ is the ball $\mathsf{ball}(\mathsf{q}, \mathsf{D}'/4)$ constructed when there is no point in $\mathsf{P} \setminus \mathsf{ball}(y, \varepsilon r_1/3)$, then it must be the case that $\mathsf{D} \leq 2\varepsilon r_1/3$ and so

$$d(\mathsf{q}, \mathsf{P}) \geq r_1/(1+\varepsilon/10) \geq \frac{3\mathsf{D}}{2\varepsilon + \varepsilon^2/5} \geq \mathsf{D}/\varepsilon.$$

It is not hard to see that in this case, y is a valid $(1+\varepsilon)$-ANN for any point inside $\mathsf{ball}(\mathsf{q}, \mathsf{D}'/4) \subseteq \mathsf{ball}(\mathsf{q}, \mathsf{D}/4)$.

(B) If the set $\mathsf{P} \setminus \mathsf{ball}(y, \varepsilon r_1/3)$ is nonempty then, as in Figure 3 let $\bar{y}$ be a $(1+\varepsilon/10)$-ANN of q in $\mathsf{P} \setminus \mathsf{ball}(y, \varepsilon r_1/3)$ and let $r_2 = d(\mathsf{q}, \bar{y})$. We divide the analysis into two cases.

(i) If $r_2 \leq 2r_1$, let $\bar{\mathsf{q}}$ be a new query point in $\mathcal{R}_\mathsf{q}$ and let $u \in \mathsf{P}$ be its nearest neighbor. If $u = y$ there is nothing to show. Otherwise, by the triangle inequality we have

$$\begin{aligned} d(\bar{\mathsf{q}}, u) &\geq d(\mathsf{q}, u) - \varepsilon r_2/5 \\ &\geq d(\mathsf{q}, y)/(1+\varepsilon/10) - \varepsilon 2r_1/5 \\ &\geq (1-\varepsilon/2) r_1. \end{aligned}$$

Again by the triangle inequality we have,

$$d(\bar{\mathsf{q}}, y) \leq d(\mathsf{q}, y) + 2\varepsilon r_1/5 = (1+2\varepsilon/5) r_1.$$

Clearly we have $d(\bar{\mathsf{q}}, y) \leq (1+\varepsilon) d(\bar{\mathsf{q}}, u)$ for $\varepsilon \leq 1/5$ and we are done.

(ii) If $r_2 > 2r_1$ then following the notation in Figure 3 we let s be the furthest point from y inside $\mathsf{ball}(y, \varepsilon r_1/3)$ and let $t = d(y, s)$. Let $\bar{\mathsf{q}}$ be a new query point and as before let $u \in \mathsf{P}$ be its nearest neighbor. We claim that the nearest neighbor of $\bar{\mathsf{q}}$ in P lies in $\mathsf{ball}(y, t)$. To see this, let z be any point in $\mathsf{P} \setminus \mathsf{ball}(y, t)$. Noting that the

distance from q to the closest point in P outside $\mathsf{ball}(y,t)$ is at least $r_2/(1+\varepsilon/10)$ and by triangle inequality,

$$\begin{aligned} d(\bar{\mathsf{q}},z) &\geq d(\mathsf{q},z) - \varepsilon r_2/5 \\ &\geq r_2/(1+\varepsilon/10) - \varepsilon r_2/5 \\ &> (1-3\varepsilon/10)r_2. \end{aligned}$$

On the other hand, we have

$$d(\bar{\mathsf{q}},y) \leq d(\mathsf{q},y) + \varepsilon r_2/5 < r_2/2 + \varepsilon r_2/5.$$

and so clearly any point in $\mathsf{P} \setminus \mathsf{ball}(y,t)$ cannot be the nearest neighbor of $\bar{\mathsf{q}}$ for $\varepsilon < 1$. Now,

$$d(\bar{\mathsf{q}},y) \leq d(\bar{\mathsf{q}},u) + t. \tag{1}$$

Now $\bar{\mathsf{q}} \in \mathsf{ball}(\mathsf{q},\varepsilon r_2/5) \setminus \mathsf{ball}(y,5t/4\varepsilon)$. We have,

$$d(\bar{\mathsf{q}},y) > 5t/4\varepsilon.$$

Then,

$$d(\bar{\mathsf{q}},u) \geq d(\bar{\mathsf{q}},y) - t \geq \left(\frac{5}{4\varepsilon} - 1\right) t. \tag{2}$$

Therefore from Equation 1 and Equation 2, we have

$$d(\bar{\mathsf{q}},y) \leq \frac{1}{1-\frac{4\varepsilon}{5}} d(\bar{\mathsf{q}},u) \leq (1+\varepsilon) d(\bar{\mathsf{q}},u)\,.$$

for $\varepsilon \leq 1/4$.

■

6.2 Bounding the number of regions created.

The online algorithm presented in Figure 3 is valid for any general metric space $\mathcal{X}$, without any restriction on the subspace of query points. However, when the query points are restricted to a subspace of low doubling dimension dim then one can show that at most $n\varepsilon^{-O(\dim)}$ regions are created. There are two types of regions created. Regions of the ***first type*** are created when $\mathsf{P} \setminus \mathsf{ball}(y,\varepsilon r_1/3)$ is empty and regions of the ***second type*** are created when this condition does not hold. An example region of the second type is shown in Figure 4. First we show that there are at most $\varepsilon^{-O(\dim)}$ regions created which are of the first type.

Lemma 6.4 *There are at most $\varepsilon^{-O(\dim)}$ regions $\mathsf{ball}(\mathsf{q},\mathsf{D}'/4)$ created by query points for which $\mathsf{P} \setminus \mathsf{ball}(y,\varepsilon r_1/3)$ is empty.*

Proof: Clearly any two such query points occur at a distance of at least $\mathsf{D}'/4$ from each other. However all of them occur inside a ball of radius $2\mathsf{D}' + 2\mathsf{D}'/\varepsilon$ around p. Thus their spread is bounded by $16(1 + 1/\varepsilon)$ and so we can have at most $\varepsilon^{-O(\dim)}$ such points. ∎

We now consider the regions of the second type created by the algorithm. Consider the mapped point set P' in the space $\mathcal{M}'$. For this we know that there is a c-WSPD $\{\{A'_1, B'_1\}, \ldots, \{A'_s, B'_s\}\}$ where c is a constant to be specified later and $s = c^{O(\dim)}n$ is the number of pairs. If a query point q creates a new region of the second type we shall assign it to the set $\mathcal{Q}'_{i,1}$ if the pair of points $y', \bar{y}'$ of the algorithm satisfy $y' \in A'_i$ and $\bar{y}' \in B'_i$. We assign it to the set $\mathcal{Q}'_{i,2}$ if $y' \in B'_i$ and $\bar{y}' \in A'_i$. For a pair $\{A'_i, B'_i\}$ of the WSPD we define the numbers $\mathsf{h}_{\max}(A'_i) = \max_{(u,h)\in A'_i} h$. Similarly let $\mathsf{h}_{\max}(B'_i) = \max_{(z,h)\in B'_i} h$ and $\mathsf{l}_i = \max_{u'\in A'_i, z'\in B'_i} d(\alpha(u), \alpha(z))$. Let $\mathsf{L}_i = \mathsf{l}_i + \mathsf{h}_{\max}(A'_i) + \mathsf{h}_{\max}(B'_i)$.

The following sequence of lemmas will then establish our claim. The basic strategy is to show that the set $\mathcal{Q}'_{i,1}$ has spread $O(1/\varepsilon^2)$. This holds analogously for $\mathcal{Q}'_{i,2}$ and so we will only work with $\mathcal{Q}'_{i,1}$. We will assume that c is a sufficiently large constant and ε is sufficiently small.

Lemma 6.5 $\mathsf{diam}\big(\mathcal{Q}'_{i,1}\big) = O(\mathsf{L}_i/\varepsilon)$.

Proof: Let q be a point in $\mathcal{Q}'_{i,1}$. By assumption we have $y' \in A'_i$ and $\bar{y}' \in B'_i$. By the triangle inequality,

$$\begin{aligned} d(y, \bar{y}) &\leq d(y, \alpha(y)) + d(\alpha(y), \alpha(\bar{y})) + d(\alpha(\bar{y}), \bar{y}) \\ &\leq \mathsf{h}_{\max}(A'_i) + \mathsf{l}_i + \mathsf{h}_{\max}(B'_i) \\ &\leq \mathsf{L}_i. \end{aligned}$$

On the other hand since the point $\bar{y}$ is outside $\mathsf{ball}(y, \varepsilon r_1/3)$ so $d(y, \bar{y}) > \varepsilon r_1/3$. This gives us $r_1 < 3\mathsf{L}_i/\varepsilon$. By Lemma 2.1 $d_{\mathcal{M}'}(\mathsf{q}', y') < 9\mathsf{L}_i/\varepsilon$. Also,

$$\begin{aligned} d_{\mathcal{M}'}(y', \bar{y}') &= d(\alpha(y), \alpha(\bar{y})) + |\mathsf{h}(y) - \mathsf{h}(\bar{y})| \\ &\leq \mathsf{l}_i + \mathsf{h}_{\max}(A'_i) + \mathsf{h}_{\max}(B'_i) \leq L_i. \end{aligned}$$

Thus let u' be any other point in A'_i (this point could be a $(1+\varepsilon/10)$-ANN found for another query point in $\mathcal{Q}_{i,1}$). By the WSPD separation property we have $d_{\mathcal{M}'}(y', u') \leq \mathsf{L}_i/c$. Thus we have

$$\begin{aligned} \mathsf{diam}\big(\mathcal{Q}'_{i,1}\big) &< 9\mathsf{L}_i/\varepsilon + \mathsf{L}_i/c + 9\mathsf{L}_i/\varepsilon \\ &= O(\mathsf{L}_i/\varepsilon), \end{aligned}$$

for ε small enough. ∎

The next lemma tells us that $r_2 = d(\mathsf{q}, \bar{y})$ is in fact $\Omega(\mathsf{L}_i)$.

Lemma 6.6 *The distances r_2 and L_i satisfy $r_2 \geq \mathsf{L}_i/18$.*

Proof: Let the point $u' \in A'_i$ attain the maximum height. So $\mathsf{h}(u) = \mathsf{h}_{\max}(A'_i)$. From the proof of Lemma 6.5,

$$d_{\mathcal{M}'}(y', u') \leq \mathsf{L}_i/c.$$

Applying the definition of the distance in $\mathcal{M}'$ this gives us,

$$\mathsf{h}_{\max}(A'_i) - \mathsf{h}(y) \leq \mathsf{L}_i/c,$$

and so $\mathsf{h}(y) \geq \mathsf{h}_{\max}(A'_i) - \mathsf{L}_i/c$. Similarly we have, $\mathsf{h}(\bar{y}) \geq \mathsf{h}_{\max}(B'_i) - \mathsf{L}_i/c$. We have $r_1 = d(\mathsf{q}, y) \geq \mathsf{h}(y)$ and also $r_2 = d(\mathsf{q}, \bar{y}) \geq \mathsf{h}(\bar{y})$. Noting that, $r_2 \geq r_1/(1 + \varepsilon/10) \geq \frac{10}{11} r_1$ we get,

$$\frac{21}{11} r_2 \geq \mathsf{h}_{\max}(A'_i) + \mathsf{h}_{\max}(B'_i) - \frac{2\mathsf{L}_i}{c}. \tag{3}$$

Let $z' \in A'_i$ and $w' \in B'_i$ be such that $d(\alpha(z), \alpha(w)) = \mathsf{l}_i$. We have $d_{\mathcal{M}'}(y', z') \leq \mathsf{L}_i/c$ and $d_{\mathcal{M}'}(\bar{y}', w') \leq \mathsf{L}_i/c$ by the WSPD separation property. Noting that $d_{\mathcal{M}'}(\mathsf{q}', y') \leq 3d(\mathsf{q}, y) \leq 3r_1$ and similarly $d_{\mathcal{M}'}(\mathsf{q}', \bar{y}') \leq 3r_2$ we have by the triangle inequality,

$$d_{\mathcal{M}'}(\mathsf{q}', z') \leq 3r_1 + \mathsf{L}_i/c,$$

and similarly,

$$d_{\mathcal{M}'}(\mathsf{q}', w') \leq 3r_2 + \mathsf{L}_i/c.$$

By the triangle inequality,

$$\begin{aligned} \mathsf{l}_i &\leq d_{\mathcal{M}'}(z', w') \\ &\leq d_{\mathcal{M}'}(z', \mathsf{q}') + d_{\mathcal{M}'}(\mathsf{q}', w') \\ &\leq 3r_1 + 3r_2 + \frac{2\mathsf{L}_i}{c} \\ &\leq \frac{63}{10} r_2 + \frac{2\mathsf{L}_i}{c}. \end{aligned}$$

Thus we have,

$$\frac{63}{10} r_2 \geq \mathsf{l}_i - \frac{2\mathsf{L}_i}{c}. \tag{4}$$

By Equation 3 and Equation 4 we have for $c \geq 8$,

$$\begin{aligned} 9r_2 &\geq \mathsf{h}_{\max}(A_i) + \mathsf{h}_{\max}(B_i) + \mathsf{l}_i - \frac{4\mathsf{L}_i}{c} \\ &\geq \mathsf{L}_i - \mathsf{L}_i/2 = \mathsf{L}_i/2, \end{aligned}$$

which immediately implies our claim. ∎

We now show that the points belonging to $\mathcal{Q}_{i,1}$ are reasonably distant from each other.

Lemma 6.7 *Let* $\mathsf{q}, \bar{\mathsf{q}} \in \mathcal{Q}_{i,1}$ *and* $\bar{\mathsf{q}}$ *comes in after* q. *Then*

$$d_{\mathcal{M}'}(\mathsf{q}', \bar{\mathsf{q}}') = d(\mathsf{q}, \bar{\mathsf{q}}) \geq \varepsilon r_2/5.$$

Proof: By Lemma 2.1 we have $d_{\mathcal{M}'}(\mathsf{q}', \bar{\mathsf{q}}') = d(\mathsf{q}, \bar{\mathsf{q}})$ and so we will only show that $d(\mathsf{q}, \bar{\mathsf{q}}) \geq \varepsilon r_2/5$.

If $\bar{\mathsf{q}} \notin \mathsf{ball}(\mathsf{q}, \varepsilon r_2/5)$ then we have nothing to prove. Otherwise, since $\bar{\mathsf{q}}$ created a new region it must be the case that $\bar{\mathsf{q}} \in \mathsf{ball}(y, 5t/4\varepsilon)$. We show that this leads to a contradiction.

The first observation is that we must have $r_2 > 2r_1/\varepsilon$. To see this notice that since $\bar{\mathsf{q}} \in \mathsf{ball}(\mathsf{q}, \varepsilon r_2/5) \cap \mathsf{ball}(y, 5t/4\varepsilon)$ it must be the case that,

$$\varepsilon r_2/5 + 5t/4\varepsilon \geq r_1.$$

But $t \leq \varepsilon r_1/3$ and so

$$\varepsilon r_2/5 + 5r_1/12 \geq r_1,$$

which gives $r_2 \geq 25r_1/12\varepsilon > 2r_1/\varepsilon$.

The next observation relates the distances L_i, r_1, r_2 and $d_{\mathcal{M}'}(y', \bar{y}')$. It is easy to see that $d_{\mathcal{M}'}(y', \bar{y}') \leq \mathsf{L}_i$. On the other hand,

$$\begin{aligned} d_{\mathcal{M}'}(y', \bar{y}') &\geq d_{\mathcal{M}'}(\mathsf{q}', \bar{y}') - d_{\mathcal{M}'}(\mathsf{q}', y') \\ &\geq r_2 - 3r_1 \\ &\geq 2r_1/\varepsilon - 3r_1 \\ &\geq r_1/\varepsilon \end{aligned}$$

for $\varepsilon \leq 1/3$.

In terms of r_2 we have

$$\begin{aligned} d_{\mathcal{M}'}(y', \bar{y}') &\geq r_2 - 3r_1 \\ &\geq r_2 - 3\varepsilon r_2 \\ &\geq r_2/2 \geq \mathsf{L}_i/36, \end{aligned}$$

by Lemma 6.6 and for $\varepsilon \leq 1/6$.

Now $\bar{\mathsf{q}}$ lies inside $\mathsf{ball}(y, 5t/4\varepsilon)$ and as $t \leq \varepsilon r_1/3$ we have $d(y, \bar{\mathsf{q}}) \leq 5r_1/12$.

Let z be an arbitrary point in B_i and notice that,

$$\begin{aligned} d_{\mathcal{M}'}(\bar{\mathsf{q}}', z') &\geq d_{\mathcal{M}'}(y', z') - d_{\mathcal{M}'}(\bar{\mathsf{q}}', y') \\ &\geq d_{\mathcal{M}'}(y', \bar{y}') - d_{\mathcal{M}'}(\bar{y}', z') - d_{\mathcal{M}'}(\bar{\mathsf{q}}', y') \\ &\geq \mathsf{L}_i/36 - \mathsf{L}_i/c - 5r_1/4 \\ &\geq \mathsf{L}_i/36 - \mathsf{L}_i/c - \frac{5\varepsilon}{4}\mathsf{L}_i \\ &\geq \mathsf{L}_i/40, \end{aligned}$$

for sufficiently small ε and sufficiently large c. This further implies that $d(\bar{\mathsf{q}}, z) \geq \mathsf{L}_i/120$.

Denote by C_i the set $A_i \cup \{s\}$ where recall that s is the furthest point from y in $\mathsf{ball}(y, \varepsilon r_1/3)$ and $d(y, s) = t$. Notice that it is possible that $s \notin A_i$. A subtle and technical point is that we require $t \neq 0$. This can be enforced by changing the definition of $\mathcal{R}_{\mathsf{q}}$ to $\mathsf{ball}(\mathsf{q}, \varepsilon r_2/5) \setminus (\mathsf{ball}(y, 4t/5\varepsilon) \setminus \{y\})$. Even with this modification, the algorithm and the

results established so far are correct. However now t cannot be 0. To see this notice that by assumption $\bar{\mathsf{q}} \in (\mathsf{ball}(y, 5t/4\varepsilon) \setminus \{y\})$. If $t = 0$ then this means the only point inside $\mathsf{ball}(y, 5t/4\varepsilon)$ is y. But $\bar{\mathsf{q}}$ cannot be y. So $t \neq 0$ and $s \neq y$. Our next observation is that $\bar{\mathsf{q}}$ is close to C_i. Let $u \in A_i$. Then,

$$\begin{aligned} d_{\mathcal{M}'}(\bar{\mathsf{q}}', u') &\leq d_{\mathcal{M}'}(\bar{\mathsf{q}}', y') + d_{\mathcal{M}'}(y', u') \\ &\leq 5r_1/4 + \mathsf{L}_i/c \\ &\leq \frac{5\varepsilon}{4}\mathsf{L}_i + \mathsf{L}_i/c, \end{aligned}$$

which also means that,

$$d(\bar{\mathsf{q}}, u) \leq \frac{5\varepsilon}{4}\mathsf{L}_i + \mathsf{L}_i/c.$$

We also have that

$$d(\bar{\mathsf{q}}, s) \leq 5r_1/12 + \varepsilon r_1/3 \leq r_1/2 \leq \varepsilon \mathsf{L}_i/2.$$

And then we observe that we can choose c large enough and ε small enough so that for any $u \in C_i$ and $z \in B_i$ we have

$$d(\bar{\mathsf{q}}, z) > 2d(\bar{\mathsf{q}}, u)\,.$$

Notice that this implies trivially that $B_i \cap C_i = \emptyset$.

Let $w \in A_i$ be the $(1+\varepsilon/10)$-ANN found by the algorithm for $\bar{\mathsf{q}}$. Since $d(\bar{\mathsf{q}}, y) \leq 5t/4\varepsilon$, it follows that $d(\bar{\mathsf{q}}, w) \leq \frac{5}{4\varepsilon}(1+\varepsilon/10)t$. Denote $d(\bar{\mathsf{q}}, w)$ by x. The next observation is that we must have $C_i \subseteq \mathsf{ball}(w, \varepsilon x/3)$. This is true because if it were not the case that C_i is entirely inside $\mathsf{ball}(w, \varepsilon x/3)$ then by the last observation, the $(1+\varepsilon/10)$-ANN of $\bar{\mathsf{q}}$ in $\mathsf{P} \setminus \mathsf{ball}(w, \varepsilon x/3)$ would belong to $C_i \setminus \mathsf{ball}(w, \varepsilon x/3)$, whereas by assumption it belongs to B_i which is disjoint from C_i.

Then we have $y, s \in \mathsf{ball}(w, \varepsilon x/3)$ and so

$$t = d(y, s) \leq 2\varepsilon d(\bar{\mathsf{q}}, w)/3 \leq \frac{2\varepsilon}{3} \cdot (1+\varepsilon/10) \cdot \frac{5t}{4\varepsilon} < t$$

for ε sufficiently small. This is a contradiction because $t > 0$. This concludes the proof. ■

The following now follows easily.

Lemma 6.8 *We have that* $\max(|\mathcal{Q}_{i,1}|, |\mathcal{Q}_{i,2}|) = \varepsilon^{-O(\dim)}$.

Proof: From Lemma 6.5, Lemma 6.6 and Lemma 6.7 it follows that the spread of the set $\mathcal{Q}'_{i,1}$ is bounded by

$$O\left(\frac{\mathsf{L}_i/\varepsilon}{\varepsilon \mathsf{L}_i}\right) = O\big(1/\varepsilon^2\big)$$

Since $\mathcal{Q}_{i,1} \subseteq \mathcal{M}'$ which is a space of doubling dimension $O(\dim)$ it follows that $|\mathcal{Q}_{i,1}| = \varepsilon^{-O(\dim)}$. The same argument works for $\mathcal{Q}_{i,2}$. ■

The next lemma bounds the number of regions created.

Lemma 6.9 *The number of regions created by the algorithm is* $n/\varepsilon^{O(\dim)}$.

Proof: As shown in Lemma 6.4 the number of regions of the first type is bounded by $\varepsilon^{-O(\dim)}$. Consider a region $\mathcal{R}_{\mathsf{q}}$ of the second type. For this point q the algorithm found a valid y and $\bar{y}$. Now from the definition of a WSPD there is some i such that $y' \in A'_i, \bar{y}' \in B'_i$ or $y' \in B'_i, \bar{y}' \in A'_i$. In other words there is some i such that $\mathsf{q} \in \mathcal{Q}_{i,1}$ or $\mathsf{q} \in \mathcal{Q}_{i,2}$. As shown in Lemma 6.8 the size of each of these is bounded by $\varepsilon^{-O(\dim)}$. Since the total number of such sets is $2s$ where $s = nc^{O(\dim)}$ is the number of pairs of the WSPD, it follows that the total number of regions created is bounded by $\left(\frac{c}{\varepsilon}\right)^{O(\dim)} n$ which is just $n\varepsilon^{-O(\dim)}$ for ε sufficiently small. ∎

We summarize the results in this section.

Theorem 6.10 *The online algorithm presented in Figure 3 always returns a* $(1+\varepsilon)$*-ANN. If the query points are constrained to a subspace of doubling dimension* dim*, then the maximum number of regions created for the online AVD by the algorithm is* $n/\varepsilon^{O(\dim)}$.

7 Conclusions

In this paper, we looked at the ANN problem when the data points can come from an arbitrary metric space (not necessarily an Euclidean space) but the query points are constrained to come from a subspace of low doubling dimension. We demonstrate that this problem is inherently low dimensional by providing fast ANN data-structures obtained by combining and extending ideas that were previously used to solve ANN for spaces with low doubling dimensions.

Interestingly, one can extend Assouad's type embedding to an embedding that $(1+\varepsilon)$-preserves distances from P to $\mathcal{M}$ (see [HM06] for an example of a similar embedding into the ℓ_∞ norm). This extension requires some work and is not completely obvious. The target dimension is roughly $1/\varepsilon^{O(\dim)}$ in this case. If one restricts oneself to the case where both P and $\mathcal{M}$ are in Euclidean space, then it seems one should be able to extend the embedding of [GK09] to get a similar result, with the target dimension having only polynomial dependency on dim. However, computing either embeddings efficiently seems quite challenging. Furthermore, even if the embedded points are given, the target dimension in both cases is quite large, and yields results that are significantly weaker than the ones presented here.

The on the fly construction of AVD without any knowledge of the query subspace (Section 6) seems like a natural candidate for a practical algorithm for ANN. Such an implementation would require an efficient way to perform point-location in the generated regions. We leave the problem of developing such a data-structure as an open question for further research. In particular, there might be a middle ground between our two ANN data-structures that yields an efficient and practical ANN data-structure while having very limited access to the query subspace.

References

[AI06] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. 47th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 459–468, 2006.

[AI08] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[AM93] P. K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:540–570, 1993.

[AM02] S. Arya and T. Malamatos. Linear-size approximate Voronoi diagrams. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 147–155, 2002.

[AMM09] S. Arya, T. Malamatos, and D. M. Mount. Space-time tradeoffs for approximate nearest neighbor searching. *J. Assoc. Comput. Mach.*, 57(1):1–54, 2009.

[AMN+98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. Assoc. Comput. Mach.*, 45(6), 1998.

[Ass83] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[CK95] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. Assoc. Comput. Mach.*, 42:67–90, 1995.

[Cla88] K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17:830–847, 1988.

[Cla06] K. L. Clarkson. Nearest-neighbor searching and metric space dimensions. In G. Shakhnarovich, T. Darrell, and P. Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Pra ctice*, pages 15–59. MIT Press, 2006.

[dBCvKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3 edition, 2008.

[GK09] L.A. Gottlieb and R. Krauthgamer. A nonlinear approach to dimension reduction. *CoRR*, abs/0907.5477, 2009.

[GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. 44th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 534–543, 2003.

[Har01] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 94–103, 2001.

[Har10] S. Har-Peled. Geometric approximation algorithms. Class notes. Online at `http://valis.cs.uiuc.edu/~sariel/teach/notes/aprx/`, 2010.

[Hei01] J. Heinonen. *Lectures on analysis on metric spaces*. Universitext. Springer-Verlag, New York, 2001.

[HKMR04] K. Hildrum, J. Kubiatowicz, S. Ma, and S. Rao. A note on the nearest neighbor in growth-restricted metrics. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms*, pages 560–561. Society for Industrial and Applied Mathematics, 2004.

[HM06] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 604–613, 1998.

[IN07] P. Indyk and A. Naor. Nearest neighbor preserving embeddings. *ACM Trans. Algo.*, 2007. To appear.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[KL04] R. Krauthgamer and J. R. Lee. Navigating nets: simple algorithms for proximity search. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 798–807. Society for Industrial and Applied Mathematics, 2004.

[KOR00] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 2(30):457–474, 2000.

[KR02] D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 741–750, 2002.

[Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. Comput.*, 106:286–303, 1993.

[MNP06] R. Motwani, A. Naor, and R. Panigrahi. Lower bounds on locality sensitive hashing. In *Proc. 22nd Annu. ACM Sympos. Comput. Geom.*, pages 154–157, New York, NY, USA, 2006. ACM.

[Tal04] K. Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 281–290, 2004.